## **JavaScript Cookies**

Cookies 用于存储 web 页面的用户信息。

## **什么是 Cookies？**

Cookies 是一些数据, 存储于你电脑上的文本文件中。

当 web 服务器向浏览器发送 web 页面时，在连接关闭后，服务端不会记录用户的信息。

Cookies 的作用就是用于解决 "如何记录客户端的用户信息":

* 当用户访问 web 页面时，他的名字可以记录在 cookie 中。
* 在用户下一次访问该页面时，可以在 cookie 中读取用户访问记录。

Cookies 以名/值对形式存储，如下所示:

username=John Doe

## **使用 JavaScript 创建Cookie**

JavaScript 可以使用 ****document.cookie**** 属性来创建 、读取、及删除 cookies。

JavaScript 中，创建 cookie 如下所示：

document.cookie="username=John Doe";

您还可以为 cookie 添加一个过期时间（以 UTC 或 GMT 时间）。默认情况下，cookie 在浏览器关闭时删除：

document.cookie="username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 GMT";

您可以使用 path 参数告诉浏览器 cookie 的路径。默认情况下，cookie 属于当前页面。

document.cookie="username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 GMT; path=/";

## **使用 JavaScript 读取 Cookie**

在 JavaScript 中, 可以使用以下代码来读取 cookies：

var x = document.cookie;

document.cookie 将以字符串的方式返回所有的 cookies，类型格式： cookie1=value; cookie2=value; cookie3=value;

## **使用 JavaScript 修改 Cookie**

在 JavaScript 中，修改 cookies 类似于创建 cookies，如下所示：

document.cookie="username=John Smith; expires=Thu, 18 Dec 2013 12:00:00 GMT; path=/";

旧的 cookie 将被覆盖。

## **使用 JavaScript 删除 Cookie**

删除 cookie 非常简单。您只需要设置 expires 参数为以前的时间即可，如下所示，设置为 Thu, 01 Jan 1970 00:00:00 GMT:

document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 GMT";

## **JavaScript Cookie 实例**

在以下实例中，我们将创建 cookie 来存储访问者名称。

首先，访问者访问 web 页面, 他将被要求填写自己的名字。该名字会存储在 cookie 中。

访问者下一次访问页面时，他会看到一个欢迎的消息。

在这个实例中我们会创建 3 个 JavaScript 函数:

1. 设置 cookie 值的函数
2. 获取 cookie 值的函数
3. 检测 cookie 值的函数

## **设置 cookie 值的函数**

首先，我们创建一个函数用于存储访问者的名字：

function setCookie(cname,cvalue,exdays)  
{  
var d = new Date();  
d.setTime(d.getTime()+(exdays\*24\*60\*60\*1000));  
var expires = "expires="+d.toGMTString();  
document.cookie = cname + "=" + cvalue + "; " + expires;  
}

****函数解析：****

以上的函数参数中，cookie 的名称为 cname，cookie 的值为 cvalue，并设置了 cookie 的过期时间 expires。

该函数设置了 cookie 名、cookie 值、cookie过期时间。

## **获取 cookie 值的函数**

然后，我们创建一个函数用户返回指定 cookie 的值：

function getCookie(cname)  
{  
var name = cname + "=";  
var ca = document.cookie.split(';');  
for(var i=0; i<ca.length; i++)   
  {  
  var c = ca[i].trim();  
  if (c.indexOf(name)==0) return c.substring(name.length,c.length);  
  }  
return "";  
}

****函数解析：****

cookie 名的参数为 cname。

创建一个文本变量用于检索指定 cookie :cname + "="。

使用分号来分割 document.cookie 字符串，并将分割后的字符串数组赋值给 ca (ca = document.cookie.split(';'))。

循环 ca 数组 (i=0;i<ca.length;i++)，然后读取数组中的每个值，并去除前后空格 (c=ca[i].trim())。

如果找到 cookie(c.indexOf(name) == 0)，返回 cookie 的值 (c.substring(name.length,c.length)。

如果没有找到 cookie, 返回 ""。

## **检测 cookie 值的函数**

最后，我们可以创建一个检测 cookie 是否创建的函数。

如果设置了 cookie，将显示一个问候信息。

如果没有设置 cookie，将会显示一个弹窗用于询问访问者的名字，并调用 setCookie 函数将访问者的名字存储 365 天：

function checkCookie()  
{  
var username=getCookie("username");  
if (username!="")  
  {  
  alert("Welcome again " + username);  
  }  
else   
  {  
  username = prompt("Please enter your name:","");  
  if (username!="" && username!=null)  
    {  
    setCookie("username",username,365);  
    }  
  }  
}